

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2001266103 A**(43) Date of publication of application: **28.09.01**

(51) Int. Cl.

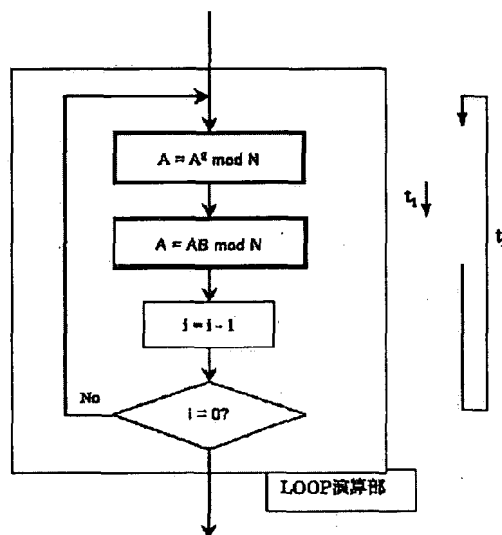
G06K 19/073**G06F 15/78****G09C 1/00**(21) Application number: **2000323178**(22) Date of filing: **23.10.00**(30) Priority: **12.01.00 JP 2000003295**(71) Applicant: **HITACHI LTD HITACHI ULSI
SYSTEMS CO LTD**(72) Inventor: **TANIMOTO CHIAKI
NAKADA KUNIIKO
NARIYOSHI YUICHIRO
TSUKAMOTO TAKU
HIRABAYASHI SHIGEO
WATASE HIROSHI
TAKAHASHI MASAOKI**(54) **IC CARD AND MICROCOMPUTER**

(57) Abstract:

PROBLEM TO BE SOLVED: To provide an IC card and a microcomputer, which reinforce security and speed up and reinforce a signal processing for security.

SOLUTION: In an IC card, an operation voltage is supplied by the electrical connection of an outer terminal with a read/write device and the input/output operation of data with a ciphering processing and a decoding processing is performed. A dummy processing operation aiming at disturbance is included in the ciphering processing or the decoding processing and the operation timing and the operation current of an inner circuit are unified. In the microcomputer of module constitution, which includes the input/output operation of data with the ciphering processing or the decoding processing, the dummy processing operation aiming at disturbance is included in the ciphering processing or the decoding processing and the operation timing and the operation current of the inner circuit are unified.

COPYRIGHT: (C)2001,JPO



文南大①

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-266103

(P2001-266103A)

(43) 公開日 平成13年9月28日 (2001.9.28)

(51) Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
G 0 6 K 19/073		G 0 6 F 15/78	5 1 0 G 5 B 0 3 5
G 0 6 F 15/78	5 1 0	G 0 9 C 1/00	6 5 0 A 5 B 0 6 2
G 0 9 C 1/00	6 5 0		6 6 0 A 5 J 1 0 4
	6 6 0	G 0 6 K 19/00	P

審査請求 未請求 請求項の数31 O L (全 26 頁)

(21) 出願番号 特願2000-323178(P2000-323178)
 (22) 出願日 平成12年10月23日 (2000.10.23)
 (31) 優先権主張番号 特願2000-3295(P2000-3295)
 (32) 優先日 平成12年1月12日 (2000.1.12)
 (33) 優先権主張国 日本 (J P)

(71) 出願人 000005108
 株式会社日立製作所
 東京都千代田区神田駿河台四丁目6番地
 (71) 出願人 000233169
 株式会社日立超エル・エス・アイ・システムズ
 東京都小平市上水本町5丁目22番1号
 (72) 発明者 谷本 千晶
 東京都小平市上水本町五丁目20番1号 株式会社日立製作所半導体グループ内
 (74) 代理人 100081938
 弁理士 徳若 光政

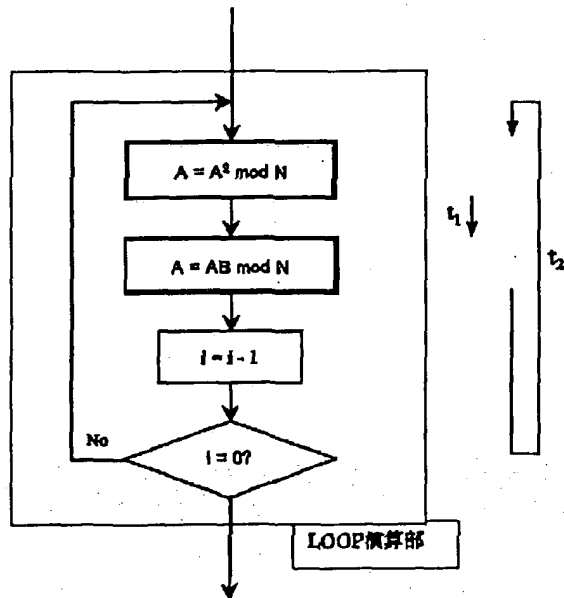
最終頁に続く

(54) 【発明の名称】 ICカードとマイクロコンピュータ

(57) 【要約】

【課題】 機密保護の強化及び機密保護のための信号処理の高速化とその強化を実現したICカードとマイクロコンピュータを提供する。

【解決手段】 外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むICカードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。



【特許請求の範囲】

【請求項1】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むICカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とするICカード。

【請求項2】 請求項1において、
上記暗号化処理又は復号化処理は、RSA暗号法などに
10 応用可能なべき乗剰乗算動作を含むものであることを特徴とするICカード。

【請求項3】 請求項2において、
上記べき乗剰乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とするICカード。

【請求項4】 請求項3において、
上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を交互に行ない、かかる演算
20 においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とするICカード。

【請求項5】 請求項4において、
上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、
30 上記演算結果は、上記Yの特定ビットe、の論理1又は0によってゲート回路を制御し、所定のレジスタに供給されるライトストロブ信号の伝達を制御して、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とするICカード。

【請求項6】 請求項4において、
上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、
40 上記演算結果は、上記Yの特定ビットe、の論理1又は0によってゲート回路を制御し、上記リードライトバッファに供給されるライトストロブ信号の伝達を制御して、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とするICカード。

【請求項7】 請求項4において、
上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブ
50

ロックであり、

上記演算結果は、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間に設けられたセクタを上記Yの特定ビットe、の論理1又は0によって制御して上記リードライトバッファに書き込まれた演算結果のうち有効なデータが所定のレジスタに格納され、無効なデータが上記ダミーレジスタに格納されるものであることを特徴とするICカード。

【請求項8】 請求項3において、
上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を交互に行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A^2 \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A^2 \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A=AB \bmod N$ の動作を継続するものであることを特徴とするICカード。

【請求項9】 請求項3において、
上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするICカード。

【請求項10】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせてなることを特徴とするICカード。

【請求項11】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードであって、

上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タ

イミング及び動作電流に不規則性を持たせてなることを特徴とするICカード。

【請求項12】 暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とするマイクロコンピュータ。

【請求項13】 請求項12において、上記モジュール構成は、1つの半導体基板上において形成されることによって実現されることを特徴とするマイクロコンピュータ。

【請求項14】 請求項13において、上記暗号化処理又は復号化処理は、RSA暗号法などに適用可能なべき乗剰余乗算動作を含み、上記べき乗剰余乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とするマイクロコンピュータ。

【請求項15】 請求項14において、上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を交互に行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項16】 請求項14において、上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を交互に行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A^2 \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A^2 \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A=AB \bmod N$ の動作を継続するものであることを特徴とするマイクロコンピュータ。

【請求項17】 請求項14において、上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$

と $A=AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項18】 請求項3において、上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、Yのビットの値に応じて、 $A=A^2 R^{-1} \bmod N$ 、 $A=AB R^{-1} \bmod N$ の演算を行うとともに、演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算 $W-N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W-N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とするICカード。

【請求項19】 請求項18において、上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、演算器のオーバーフローフラグOVの有無が記憶され、上記剰余乗算の後に上記第1記憶回路の演算結果WからNの減算 $W-N$ が行われて、その演算結果が上記オーバーフローフラグOVが有る時には上記第1の記憶回路に格納され、オーバーフローフラグOVが無い時には上記第1記憶回路とは異なる第2の記憶回路に上記錯乱目的のダミー動作として書き込まれ、上記第1の記憶回路の演算結果が有効なデータとして出力されることを特徴とするICカード。

【請求項20】 請求項18において、上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、演算器のオーバーフローフラグOVの有無が記憶され、上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算 $W-N$ が行われて、オーバーフローフラグOVが有るときに上記演算結果 $W-N$ がセクタにより選択され、オーバーフローフラグOVが無いときには上記第1記憶回路の演算結果Wがセクタにより選択されて第2の記憶回路に格納されて有効なデータとして出力されることを特徴とするICカード。

【請求項21】 請求項18において、上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、演算器のオーバーフローフラグOVの有無が記憶され、上記剰余乗算の後に上記第1の記憶回路の演算結果Wか

らNの減算 $W-N$ が行われ、オーバーフローフラグOVが有るときには減算 $W-N$ が第2の記憶回路に記憶され、オーバーフローフラグOVが無いときには減算 $W-N$ が第3の記憶回路に記憶され、オーバーフローフラグOVが有るときには上記第2の記憶回路のデータが有効なデータとして出力され、オーバーフローフラグOVが無いときには上記第1の記憶回路のデータが有効なデータとして出力されること特徴とするICカード。

【請求項22】 請求項18において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、
演算器のオーバーフローフラグOVの有無が記憶され、
上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算結果 $W-N$ が第2の記憶回路に格納され、オーバーフローフラグOVが無いとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、
上記第2の記憶回路を選択するアドレスにより第1の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグOVが有るとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして第2の記憶回路の演算結果を有効なデータとして出力させることを特徴とするICカード。

【請求項23】 請求項18において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、
演算器のオーバーフローフラグOVの有無が記憶され、
上記剰余乗算の後に上記第1の記憶回路と第2の記憶回路のアドレスが交換され、第2の記憶回路を選択するアドレスにより選択される演算結果値WからNの減算 $W-N$ が行われて第1の記憶回路を選択するアドレスにより選択される第2の記憶回路に減算結果 $W-N$ が格納され、オーバーフローフラグOVが有るときにのみ上記アドレスを再度交換し、第1の記憶回路を選択するアドレスにより選択される第1又は第2の記憶回路のデータを有効なデータとして出力させることを特徴とするICカード。

【請求項24】 請求項18において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、
上記剰余乗算の後に上記第1の記憶回路の演算結果値WからNの減算 $W-N$ が行われて第2の記憶回路に格納され、
この $W-N$ の減算が行われた時の演算器からボローフラグBRが記憶され、
ボローフラグBRが有るときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、
上記第2の記憶回路を選択するアドレスにより第1の記憶回路の演算結果Wを出力し、
ボローフラグBRが無いときには、第1の記憶回路と第

2の記憶回路を選択する最下位アドレスをそのままし、
て、上記第2の記憶回路を選択するアドレスにより第2の記憶回路の演算結果 $W-N$ を出力させることを特徴とするICカード。

【請求項25】 請求項14において、
上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、Yのビットの値に応じて、 $A=A^2 R^{-1} \bmod N$ 、 $A=ABR^{-1} \bmod N$ の演算を行うとともに、

10 演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算 $W-N$ を行なう正規動作と、
各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W-N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、

上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とするマイクロコンピュータ。

【請求項26】 請求項25において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、
20 演算器からのオーバーフローフラグOVの有無が記憶され、

上記剰余乗算の後に上記第1記憶回路の演算結果WからNの減算 $W-N$ が行われて、その演算結果が上記オーバーフローフラグOVが有る時には上記第1の記憶回路に格納され、オーバーフローフラグOVが無い時には上記第1記憶回路とは異なる第2の記憶回路に上記錯乱目的のダミー動作として書き込まれ、
上記第1の記憶回路の演算結果が有効なデータとして出力されることを特徴とするマイクロコンピュータ。

30 【請求項27】 請求項25において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、

演算器のオーバーフローフラグOVの有無が記憶され、
上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算 $W-N$ が行われて、オーバーフローフラグOVが有るときに上記演算結果 $W-N$ がセレクトタにより選択され、オーバーフローフラグOVが無いときには上記第1記憶回路の演算結果Wがセレクトタにより選択されて第2の記憶回路に格納されて有効なデータとして出力されること特徴とするマイクロコンピュータ。

40 【請求項28】 請求項25において、
上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、
演算器のオーバーフローフラグOVの有無が記憶され、
上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算 $W-N$ が行われ、オーバーフローフラグOVが有るときには減算結果 $W-N$ が第2の記憶回路に記憶され、オーバーフローフラグOVが無いときには減算結果 $W-N$ が第3の記憶回路に記憶され、

50 オーバーフローフラグOVが有るときには上記第2の記

憶回路のデータが有効なデータとして出力され、オーバーフローフラグOVが無いときには上記第1の記憶回路のデータが有効なデータとして出力されること特徴とするマイクロコンピュータ。

【請求項29】 請求項25において、

上記 $A^2 \cdot R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、

演算器のオーバーフローフラグOVの有無が記憶され、上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算結果 $W-N$ が第2の記憶回路に格納され、オーバーフローフラグOVが無いとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するアドレスにより第1の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグOVが有るとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして第2の記憶回路の演算結果を有効なデータとして出力させることを特徴とするマイクロコンピュータ。

【請求項30】 請求項25において、

上記 $A^2 \cdot R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、

演算器のオーバーフローフラグOVの有無が記憶され、上記剰余乗算の後に上記第1の記憶回路と第2の記憶回路のアドレスが交換され、第2の記憶回路を選択するアドレスにより選択される演算結果値WからNの減算 $W-N$ が行われて第1の記憶回路を選択するアドレスにより選択される第2の記憶回路に減算結果 $W-N$ が格納され、オーバーフローフラグOVが有るときにのみ上記アドレスを再度交換し、第1の記憶回路を選択するアドレスにより選択される第1又は第2の記憶回路のデータを有効なデータとして出力させることを特徴とするマイクロコンピュータ。

【請求項31】 請求項25において、

上記 $A^2 \cdot R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納され、

上記剰余乗算の後に上記第1の記憶回路の演算結果値WからNの減算 $W-N$ が行われて第2の記憶回路に格納され、

この $W-N$ の減算が行われた時の演算器からボローフラグBRが記憶され、

ボローフラグBRが有るときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するアドレスにより第1の記憶回路の演算結果Wを出力し、

ボローフラグBRが無いときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして、上記第2の記憶回路を選択するアドレスにより第2の記憶回路の演算結果 $W-N$ を出力させることを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、ICカードとマイクロコンピュータに関し、特にICカードやプログラム内蔵の1チップマイクロコンピュータのようなCPUとメモリを含み暗号鍵を使ったデータ処理を行なうものの機密保護技術に利用して有効な技術に関するものである。

【0002】

【従来の技術】メモリに保存されている鍵情報を用いてデータの暗号処理化又は復号化処理を行なうようにしたICカードにおいて、処理時間の違いを利用して実行内容や暗号鍵を推定するTA (Timing Attack) 法のようなハッキング手法に対抗するため、暗号処理化又は復号化処理の実行中又は実行の前後に、鍵情報の内容との時間的な相関関係を喪失させる遅延処理を実行する技術の例として、特開平10-69222号がある。また、ICカードに関しては、オーム社出版電子情報通信学会編水沢順一著「ICカード」などがある。

【0003】

【発明が解決しようとする課題】近年、ICカードが暗号処理を行っている時の消費電流を観測して解析することにより、容易に暗号処理の内容や暗号鍵が推定されることの可能性が示唆されている。このことについては、John Wiley & sons 社 W.Rankl & W. Effing著「Smart Card Handbook」の8.5.1.1 Passive protective mechanisms (263ページ) に記載されている。

【0004】つまり、SPA (Simple Power Analysis) 法では、演算命令の違い、あるいは処理されているデータの違いにより生じる消費電流波形の違いから、暗号鍵や処理されているデータを解析し、DPA (Differential Power Analysis) 法では、消費電流波形を統計処理して暗号鍵を推定する。このDPA法では、例えばDESのある部分に仮定した暗号鍵をあてはめて、平文を変化させながら消費電流波形を測定して統計する。暗号鍵を様々に変化させながらこの作業を繰り返し、正しい鍵のときには電流波形が大きなピークを示す。

【0005】前記公報に記載のようにTA (Timing Attack) 法のみを考慮した遅延処理では、実際の演算による消費電流の相関性までも喪失させることができず、上記のような消費電流波形を観測するというSPA又はDPA法のようなハッキング手法には対抗できない。そこで、本願発明者等においては、上記ICカード及びICカード等のようなモジュールに搭載されるマイクロコンピュータのように内蔵のプログラムにより一定のデータ処理動作を行うものに対して上記のような消費電流の観測による暗号処理の内容や暗号鍵の解読をより確実に防止することができる機密保護技術を開発するに至った。

【0006】この発明の目的は、機密保護の強化を実現したICカードとマイクロコンピュータを提供することにある。この発明の他の目的は、機密保護のための信号

処理の高速化とその強化を実現したICカードとマイクロコンピュータを提供することにある。この発明の前記ならびにそのほかの目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【0007】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むICカードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た擬乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【0008】本願において開示される発明のうち他の代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た擬乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【0009】

【発明の実施の形態】図1には、この発明が適用されるICカードの一実施例の外観図が示されている。ICカードは、プラスチックケースからなるカード101と、かかるカード101の内部に搭載された図示しないチップのマイクロコンピュータ等からなるICカード用チップを持つものである。上記ICカードは、さらに上記ICカード用チップの外部端子に接続されている複数の接点（電極）102を持つ。複数の接点102は、後で図2によって説明するような電源端子VCC、電源基準電位端子VSS、リセット入力端子RESバー、クロック端子CLK、データ端子I/O-1/I/RQバー、I/O-2/I/RQバーとされる。ICカードは、かかる接点102を通して図示しないリーダーライタのような外部結合装置から電源供給を受け、また外部結合装置との間でのデータの通信を行う。

【0010】図2には、この発明に係るICカードに搭載されるICカード用チップ（マイクロコンピュータ）の一実施例の概略ブロック図が示されている。同図の各回路ブロックは、公知のMOS集積回路の製造技術により、特に制限されないが、単結晶シリコンのような1個の半導体基板上において形成される。

【0011】この発明に係るICカード用チップの構成は、基本的にマイクロコンピュータと同じような構成である。その構成は、クロック生成回路205、中央処理装置（以下単にCPUという場合がある）201、ROM(Read Only Memory)206やRAM(Random Access Memory)207、不揮発性メモリ208などの記憶装置、

暗号化及び復号化処理の演算を行なうコプロセッサ209、入出力ポート（I/Oポート）202などからなる。

【0012】クロック生成回路205は、図示しないリーダーライタ（外部結合装置）から図1の接点102を介して供給される外部クロックCLKを受け、かかる外部クロック信号に同期したシステムクロック信号を形成し、それをチップ内部に供給する回路である。CPU201は、論理演算や算術演算などを行う装置であり、システムコントロールロジック、乱数発生器及びセキュリティロジック及びタイマなどを制御する。記憶装置206、207、208は、プログラムやデータを格納する装置である。コプロセッサ209は、後述するようにRSA暗号法などに応用可能なべき乗剰余乗算動作を行なう演算器とレジスタ及び制御論理から構成される。I/O（入出力）ポート202は、リーダーライタと通信を行う装置である。データバス204とアドレスバス203は、各装置を相互に接続するバスである。

【0013】上記記憶装置206、207、208のうち、ROM206は、記憶内容が不揮発的に固定されているメモリであり、主にプログラムを格納するメモリである。揮発性メモリ（以下、RAMという）207は自由に記憶情報の書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容が消えてなくなる。ICカードがリーダーライタから抜かれると電源の供給が中断されるため、RAM207の内容は、保持されなくなる。

【0014】上記不揮発性メモリ（以下、EEPROM(Electrical Erasable Programmable Read Only Memory)という）208は、内容の書き換えが可能な不揮発性メモリであり、その中に一旦書き込まれた情報は、電源の供給が停止されてもその内部に保持される。このEEPROM208は、書き換える必要があり、かつICカードがリーダーライタから抜かれても保持すべきデータを格納するために使われる。例えば、ICカードがブリベイドカードとして使用されるような場合、のブリベイドの度数などは、使用するたびに書き換えられる。この場合の度数などは、リーダーライタが抜かれてもICカード内で記憶保持する必要があるため、EEPROM208で保持される。

【0015】CPU201は、いわゆるマイクロプロセッサと同様な構成にされる。すなわち、その詳細を図示しないけれども、その内部に命令レジスタ、命令レジスタに書込まれた命令をデコードし、各種のマイクロ命令ないしは制御信号を形成するマイクロ命令ROM、演算回路、汎用レジスタ（RG6等）、内部バスBUSに結合するバスドライバ、バスレシーバなどの入出力回路を持つ。CPU201は、ROM206などに格納されている命令を読み出し、その命令に対応する動作を行う。CPU201は、I/Oポート202を介して入力され

る外部データの取り込み、ROM206からの命令や命令実行のために必要となる固定データのようなデータの読み出し、RAM207やEEPROM208に対するデータの書き込みと読み出し動作制御等を行う。

【0016】上記CPU201は、クロック生成回路205から発生されるシステムクロック信号を受けそのシステムクロック信号によって決められる動作タイミング、周期をもって動作される。CPU201は、その内部の主要部がPチャンネル型MOSFETとNチャンネル型MOSFETとからなるCMOS回路から構成される。特に制限されないが、CPU201は、CMOSスタティックフリップフロップのようなスタティック動作可能なCMOSスタティック回路と、信号出力ノードへの電荷のプリチャージと信号出力ノードへの信号出力とをシステムクロック信号に同期して行うようなCMOSダイナミック回路とを含む。

【0017】ICカードのセキュリティ機能としては、チップ内部で乱数を自動生成する乱数発生器や、ランダムに割込みを生成するタイマー機能などの他に、本願発明にかかる高セキュリティ機能として、ICカードと外部装置とのデータ送受信の際に用いるRSA暗号法などに应用可能なべき乗剰余演算動作を行なう暗号処理用演算ユニット(コプロセッサ)209を内蔵している。このコプロセッサ209は専用のレジスタが内蔵されている。

【0018】ICカードにおけるセキュリティ・システムでは、通信データの暗号処理は必須であり、この実施例でも現在最も多く利用されている公開鍵暗号としてRSA暗号が用いられる。この暗号法では、暗号化・復号化とともにべき乗剰余乗算 $X^i \bmod N$ を用いるが、これは古くから知られている計算アルゴリズムによって剰余乗算 $A^i \bmod N$ と $AB \bmod N$ の2つの形に分解することができる。つまり、 $Y = e_n, e_{n-1}, \dots, e_1$ の値 e_i を上位 e_n から最下位の e_1 まで順に1ビットずつ見ていき、 $e_i = 0$ だったら $A^i \bmod N$ のみを、 $e_i = 1$ だったら $A^i \bmod N$ と $AB \bmod N$ を演算する。したがって、 $e_i = 0$ のときには $A^i \bmod N$ の演算の後に $i = 0$ であるかの判定処理が行なわれ、 $e_i = 1$ のときには $A^i \bmod N$ と $AB \bmod N$ との演算の後に $i = 0$ であるかの判定処理が行なわれるために、 $e_i = 0$ と1とに対応した2通りの電流波形の形態が現れてしまう。

【0019】この実施例のようにコプロセッサ209を用いた場合には、その消費電流はCPUの消費電流に比べて比較的大きいため、この部分の電流波形を観測することによりコプロセッサの動作形態を比較的容易に識別することができ、前記TA法とSPA法により暗号鍵Yの値を解読されてしまう可能性が高い。そこで、この実施例のコプロセッサ209では、上記暗号化・復号化とともに用いられるべき乗剰余乗算 $X^i \bmod N$ の演算を行

なうに当たり攪乱目的のダミーの演算が挿入される。つまり、図3のタイミング図及び図4のフローチャート図に示すように $e_i = 0$ でも1でも $A^i \bmod N$ と $AB \bmod N$ の両方の演算を常に行なうようにするものである。

【0020】図3のタイミング図において、(a)に示すように本来は、 $e_n = 1$ のときには $A^i \bmod N$ の演算を行い、 e_n の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント($n-1$)して $i = 0$ の判定に時間 t_2 を費やす。次いで、次ビット $e_{n-1} = 0$ のときは、 $A^i \bmod N$ の演算を行い、 $e_{n-1} = 0$ の判定と i をデクリメント($n-2$)して $i = 0$ の判定に時間 t_3 を費やす。そして、次ビット $e_{n-2} = 1$ のときには、 $A^i \bmod N$ の演算を行い、 e_{n-2} の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント($n-3$)して $i = 0$ の判定に時間 t_2 を費やす。以下、同様に e_i まで同様な動作を繰り返すものである。

【0021】この実施例のコプロセッサ209においては、上記暗号鍵Yの各ビット e_i の論理0又は1に無関係に $A^i \bmod N$ の演算の後に $AB \bmod N$ の演算を行なうようにする。図3(b)の $e_{n-1} = 0$ のときのように e_i が論理0のときにおける上記 $AB \bmod N$ の演算が攪乱目的のダミー演算として挿入される。つまり、

(b)のタイミング図及び図4のフローチャート図のように、 $A^i \bmod N$ と $AB \bmod N$ の演算動作の間には、例えば e_i の判定の判定を含む時間 t_1 が費やされ、 $AB \bmod N$ と次ビットに対応した $A^i \bmod N$ の演算動作の間には、 i のデクリメント動作と $i = 0$ の判定時間 t_2 が費やされる画一化された動作タイミング及び動作電流とすることができる。ただし、この実施例では、 e_i の判定処理は、その結果が演算動作の分岐の条件とされないため図4のフローチャート図では省略されている。

【0022】図5には、上記コプロセッサの一実施例のブロック図が示されている。この実施例では、主に演算器、制御論理、専用レジスタブロックより構成され、べき乗剰余演算の最終結果はデータバッファ、データバスを介して中央処理装置CPUに送信される。専用レジスタは、アドレスバスから供給されるアドレス信号に対してその選択動作が行なわれる。

【0023】この実施例では、内部バスMDBとレジスタブロックのリードライトバッファ(R/W Buffer)との間にゲート回路1が設けられる。このゲート回路1は、制御論理により制御が行なわれて、 e_i が論理0ならば $A^i \bmod N$ 動作の演算結果が内部バスMDBとリードライトバッファを介して所定のレジスタCDAに取り込まれた後開いていたゲートが閉じるようにされる。つまり、上記演算結果がリードライトバッファに取り込まれると、その後にゲートを閉じてしまいリードライ

バッファへの新たなデータの書き込みを禁止する。したがって、その後に行なわれる $AB \bmod N$ の演算結果は無効データとして扱われることとなる。また、 e_i が論理1ならばゲート回路1はゲートを開いた状態のままとされる。

【0024】図6には、上記コプロセッサの他の一実施例のブロック図が示されている。この実施例では、レジスタブロックのリードライトバッファ(R/W Buffer)と各レジスタとの間にゲート回路2が設けられる。このゲート回路2は、前記同様に制御論理により制御が行なわれて、 e_i が論理0ならば $A^i \bmod N$ 動作の演算結果が内部バスMDBとリードライトバッファとを介して所定のレジスタCDAに書き込まれた後に開いていたゲートが閉じるようにされる。つまり、上記演算結果がレジスタCDAに取り込まれると、その後ゲートを閉じてしまいかかるレジスタCDAへの新たなデータの書き込みを禁止する。したがって、その後に行なわれる $AB \bmod N$ の演算結果は、リードライトバッファまでは書き込まれるが、実際には無効データとして扱われることとなる。また、 e_i が論理1ならばゲート回路2はゲートを開いた状態のままとされる。

【0025】図7には、上記ゲート回路の一実施例の内部構成図が示されている。ダミー書き込み制御ユニットは、アンドゲート回路によって構成され、一方の入力には制御論理からのライトイネーブル信号が供給され、他方の入力には演算器で生成されたライトストロープ信号が供給される。上記ゲート回路の出力信号は、データバッファ(R/W Buffer)と専用レジスタにライトストロープ信号として伝えられる。

【0026】この実施例では、演算結果そのものの伝達制御するものに代えて、レジスタ又はデータバッファへの書き込み動作を指示するライトストロープ信号の発生タイミングを切り換えるようにするものである。つまり、 $e_i = 0$ のときには、 $A^i \bmod N$ 動作の演算結果が出力された後にライトイネーブル信号をロウレベルとしてアンドゲート回路のゲートが閉じるようにするものである。逆に、 $e_i = 1$ のときには、制御論理はライトイネーブル信号をハイレベルのままとして、演算器で形成されたライトストロープ信号がそのままデータバッファ又は専用レジスタに伝えられる。この構成では、複数ビットからなる演算結果Aに対応して、複数個のゲート回路を設ける必要がないので簡素化が可能になる。

【0027】図8には、上記コプロセッサの他の一実施例のブロック図が示されている。この実施例では、レジスタブロックのリードライトバッファ(R/W Buffer)と各レジスタとの間にセクタ2とレジスタブロックにダミーレジスタ1が設けられる。このセクタ2は、前記同様に制御論理により制御が行なわれて、 e_i が論理0ならば $A^i \bmod N$ 動作の演算結果が内部バスMDBとリードライトバッファとを介して所定のレジスタC

DAに書き込まれるような信号経路を形成し、その後ダミーレジスタ1を選択するような信号経路を形成する。

【0028】つまり、上記演算結果がレジスタCDAに取り込まれると、その後ダミーレジスタ1を選択するので、レジスタCDAへの新たなデータの書き込みを禁止しつつその後に行なわれる $AB \bmod N$ の演算結果がダミーレジスタに書き込まれものとなる。 e_i が論理1ならばセクタ2は常にレジスタCDAを選択する。この構成は、演算結果をレジスタに書き込む動作を含めて e_i が論理0のときと論理1のときとで電流波形でみたときに全く同一にすることができるから、電流波形を利用したアタックをより確実に無力化することができる。

【0029】図9には、この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図が示されている。図9(a)のタイミング図及び(b)のフローチャート図において、前記説明したように、 $A^i \bmod N$ の演算後、 e_i の判定の時間 t_1 の間ダミー演算動作として $A^i \bmod N$ を継続して $AB \bmod N$ の演算に移行する。

【0030】その演算後に i をデクリメント(-1)して $i = 0$ の判定に時間 t_2 を費やすが、その間も上記 $AB \bmod N$ の演算を継続させる。以下、同様に e_i まで同様な動作を繰り返すものである。この構成は、演算動作中は、 e_i が論理0と1のときに関係なく上記のような演算動作を継続するので、電流波形でみたときに格別な特徴を見出すことができないから、電流波形を利用したアタックを無力化することができる。

【0031】図10には、図9のコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーイネーブル信号とコプロイネーブル信号を送出する。上記ダミーイネーブル信号とコプロイネーブル信号は、オアゲート回路を通して演算器に入力される。それ故、コプロイネーブル信号がアクティブであるときに加えて、ダミーイネーブル信号がアクティブであるときにも演算器は演算動作を行なうようにされる。

【0032】上記ダミーイネーブル信号は、インバータ回路を通してアンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力には演算器で形成されたライトストロープ信号が供給される。つまり、演算器で形成されたライトストロープ信号の伝達をダミーイネーブル信号で選択的に停止できるようにする。コプロイネーブル信号がアクティブにされて、前記正規の演算動作が終了すると、その演算結果を出力するためのライトストロープ信号が形成される。このようにコプロイネーブル信号がアクティブのときには、ダミーイネーブル信号の反転信号がアクティブレベルとなってアンドゲート回路のゲートを開くように制御するので、上記正規演算結果はライトストロープ信号によって、R

／Wバッファ又はレジスタブロックの所定のレジスタに書き込まれる。

【0033】上記のような正規演算が終了すると、ダミーイネーブル信号がアクティブとなって演算器に対して演算動作を指示する。この演算の終了によって、上記ライトストロープ信号が形成されるが、上記ダミーイネーブル信号の反転信号によってアンドゲート回路がゲートを閉じているので、上記攪乱目的のダミー演算動作によって発生されたライトストロープ信号がR／Wバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミー演算結果は無効データとして消失させられる。

【0034】図11には、この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図が示されている。前記図3に示した実施例のように、攪乱目的のダミー演算を挿入して、(a)のタイミング図のように、 e_i に対して画一化して $A^i \bmod N$ と $AB \bmod N$ の演算を一対として行なうようにした場合でも、各演算には、演算結果にオーバーフロー処理を必要とするもの(あり)のものと、オーバーフロー処理を必要としないもの(なし)が発生する。

【0035】このようなオーバーフロー処理は、演算時間を長くするものであるため電流波形でみると、オーバーフロー処理ありとなしとの識別が可能になる。このような電流波形の特徴から演算内容や演算データを推測することも不可能ではないと考えられるため、この実施例では(b)のタイミング図に示すようにオーバーフロー処理を不要とする演算に対しても必要なときと同様にオーバーフロー処理を挿入する。つまり、みかけ上は、全ての演算 $A^i \bmod N$ と $AB \bmod N$ の演算において画一的にオーバーフロー処理のための動作が実施されるために、その識別を無力化するものである。

【0036】図12は、この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図が示されている。このフローチャート図は、前記図11(b)に対応している。 $A^i \bmod N$ と $AB \bmod N$ の各演算は、剰余演算部とオーバーフロー演算部からなり、演算結果に無関係に上記オーバーフロー演算処理を実施するものである。

【0037】図13には、この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図が示されている。この実施例による対策前では、前記 $A^i \bmod N$ と $AB \bmod N$ のようなコプロ演算においては、その演算結果に対応してオーバーフロー処理のあるものと無いもの2種類が存在したが、この実施例による対策後では、前記 $A^i \bmod N$ と $AB \bmod N$ のようなコプロ演算においては、その演算結果に無関係に常にオーバーフロー処理が実行される。このため、本来はオーバーフロー処理が不要な演算動作に対して実施されたオーバーフロー処理は、攪乱目的のダミー動作とされ

る。

【0038】図14には、図11ないし図13に示したコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーオーバーフロー信号とコプロオーバーフロー信号を送出する。上記ダミーオーバーフロー信号とコプロオーバーフロー信号は、オアゲート回路を通して演算器に入力される。それ故、コプロオーバーフロー信号がアクティブであるときに加えて、ダミーオーバーフロー信号がアクティブであるときにも演算器はオーバーフロー処理動作を行なうようにされる。

【0039】上記コプロオーバーフロー信号は、アンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力に演算器で形成されたライトストロープ信号が供給される。つまり、演算器で形成されたライトストロープ信号の伝達をコプロオーバーフロー信号がアクティブレベルでないときに選択的に停止できるようにする。つまり、コプロオーバーフロー信号がアクティブレベルでないときはダミーオーバーフロー信号によって演算器がオーバーフロー処理を行なっているため、かかるオーバーフロー処理で形成されたライトストロープ信号は上記ゲート回路のゲートを閉じることによって無効にするものである。したがって、前記正規のオーバーフロー処理終了すると、その処理結果を出力するためのライトストロープ信号が形成されて、R／Wバッファ又はレジスタブロックの所定のレジスタに処理結果が書き込まれる。

【0040】これに対して、ダミーオーバーフロー信号がアクティブとなって演算器に対してオーバーフロー処理動作を指示した場合には、そのオーバーフロー処理によって形成されたライトストロープ信号は、上記コプロオーバーフロー信号によってアンドゲート回路のゲートが閉じられるものであるから、上記攪乱目的のダミーオーバーフロー処理動作によって発生されたライトストロープ信号がR／Wバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミーオーバーフロー処理結果は無効データとして消失させられる。

【0041】図15には、この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図が示されている。(a)に示すように本来は、 $e_n = 1$ のときには $A^i \bmod N$ の演算を行い、 e_n の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント($n-1$)して $i=0$ の判定に時間 t_2 を費やす。次いで、次ビット $e_{n-1}=0$ のときは、 $A^i \bmod N$ の演算を行い、 $e_{n-1}=0$ の判定と i をデクリメント($n-2$)して $i=0$ の判定に時間 t_3 を費やすような演算動作に対して、上記各演算毎の時間 t_1 、 t_2 及び t_3 に対して攪乱目的のダミーサイクルが挿入される。

【0042】(b)のタイミング図では、上記攪乱目的のダミーサイクルの挿入は、各演算毎の時間を最も長い時間 t_3 に揃えるように挿入するものである。これにより、時間 t_3 をインターバルとして $A^2 \bmod N$ 又は $AB \bmod N$ のいずれかの演算が実施されるために、みかけ上は演算動作に対応した電流波形が画一化されてその識別を無力化するものである。これに対して、(c)タイミング図では、上記(b)とは逆に上記演算毎のインターバルにおいて時間がランダムに変化する攪乱目的のダミーサイクルが挿入される。上記 $A^2 \bmod N$ 又は $AB \bmod N$ のいずれかの演算が時間的にランダムに実施される。そのため、電流波形でみると上記各演算動作と無関係で、かつ不規則性の電流値にされる。言い換えるならば、上記演算器において同じ状態及び同じ動作でも毎回異なるよう、統計的な観点での非再現性を持つようにされるために、その識別を無力化することができる。

【0043】上記のような攪乱目的のダミーサイクルは、前記図2に示されたようにタイマーを利用して演算間隔を可変にするものである。あるいはコプロセッサの外部にタイマーを設けて一定の時間が経過するまで次の演算の実行を待つようにする。つまり、コプロセッサによるべき乗剰余乗算の演算において、図15(a)に示した前記各演算毎の時間 t_1 、 t_2 、 t_3 に攪乱目的のダミーのサイクルを挿入し、一定時間後にタイマーからの割込みを入れる。これにより、図15(b)に示すように t_1 、 t_2 、 t_3 の時間が全て一定となり、電流波形からのアタックを困難にする。あるいはタイマーには乱数発生器で生成した乱数をセットしておき、(c)に示すように毎回 t_1 、 t_2 、 t_3 の時間をランダムに変化させることも可能である。また、タイマーを用いなくとも、ソフトウェアでカウントすることも可能である。

【0044】べき乗剰余乗算において、コプロセッサによる演算の高速化を目的とし、Yの値を2ビット、あるいは3ビットずつ処理するようにすると、例えば図16のフローチャート図に示すように、2ビット処理の例で説明するなら常に $A^2 \bmod N - A^2 \bmod N - AB \bmod N$ 及び $i-2$ と $i=0?$ の各ステップの繰り返しになるので、前記1ビットずつ行なう場合のような攪乱目的のダミー演算を行なわなくとも、処理時間や電流波形が一定になる。そのため、電流波形からYの値を推定するのは困難になる。また演算の回数も、前記のバイナリ法だと最大で2n回かかっていたものを、2ビット処理だと常に1.5n回で済むために、動作時間の短縮にもつながる。

*

制御用レジスタ (CCNT)

ビット7	ビット6	ビット2	ビット1	ビット0
—	—			e_1	e_{i-1}

*【0045】コプロセッサの演算が開始するまでにA、B、Nの値をそれぞれコプロセッサ専用レジスタに転送し格納しておく。しかしながら、2ビット処理を行う場合、Yの値によって4通りのBの値 B_1 、 B_2 、 B_3 、 B_4 が必要になり、これらの値は前もって計算して、RAMやEEPROMなどに格納しておき、毎回コプロセッサ専用レジスタに転送することになるが。この際、4通りのBの値によって転送中の電流波形に特徴が現れる可能性がある。

10 【0046】例えば、16ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前にすべてのバスの値を“1”にそろえるバスである。このバスに、値は違おうが“1”のビットの数が同じデータ、例えば、“1”のビットの数が2である16進数で“88”と“11”、を転送した場合、電流波形はほぼ同じ波形になると予測される。この理由は、“1”から“0”へ変化したビットの数が同じであるため、同じように電流を消費し、同じ電流波形になるからである。

20 【0047】もし、“1”のビットの数が1つ異なるデータ、例えば、“1”のビットの数が3である“89”や“19”を転送した場合、“1”のビットの数が2のデータとは消費電流が異なる。これは、13ビット分バスの値が“1”から“0”に変わったため、その分の電流が消費される。そのため、先の14ビットが変化したデータに比べて消費電流が1ビット分小さくなる。一般に、変化するビットの数が多ほど電流波形は高くなるという規則性がある。この規則性から転送されているデータを推定することができると思われる、電流アタックの対象となりやすい。これを防ぐため次のような工夫を行なうものである。

30 【0048】図17と図18には、この発明に係るコプロセッサの他の一実施例のブロック図がそれぞれ示されている。この実施例のコプロセッサは、2ビット処理と3ビット処理に向けられている。つまり、コプロセッサのレジスタ容量を増やして、2ビット処理の場合には4通りのBの値 $B_1 \sim B_4$ を、3ビット処理の場合には8通りのBの値 $B_1 \sim B_8$ をコプロセッサのレジスタに格納しておく。従って、演算の途中で記憶回路(RAM)からデータバスを通して上記コプロセッサのレジスタに前記のような転送の必要がなくなり、前記電流アタックに対して防御することができる。

【0049】

ビット2	e_i	e_{i-1}	演算の種類
0	0	0	$A \leftarrow A^2 \bmod N$
0	1	0	$A \leftarrow A \bmod N$
0	1	1	$A \leftarrow A \times N$
1	0	0	$A \leftarrow AB_1 \bmod N$
1	0	1	$A \leftarrow AB_2 \bmod N$
1	1	0	$A \leftarrow AB_3 \bmod N$
1	1	1	$A \leftarrow AB_4 \bmod N$

【0051】つまり、前記図16に示したようなフローチャート図において、コプロセッサが $AB \bmod N$ を実行する際、下記のように4つ（3ビット処理のときにはあるいは8つ）のうちの正しいBレジスタCDBから値を選んで実行できるように、Yの2ビット（あるいは3ビット）の値をコプロセッサの制御レジスタ（CCNT）のビットに当てはめ、前記に示す制御レジスタ及び演算の種類のように、2ビット処理の場合には、 $AB_1 \bmod N$ 、 $AB_2 \bmod N$ 、 $AB_3 \bmod N$ 、 $AB_4 \bmod N$ のうちのどの演算をするかを選択させるようにする。

【0052】図19には、この発明に係るコプロセッサの他の一実施例のブロック図が示されている。この実施例のコプロセッサも、2ビット処理や3ビット処理のような複数ビット処理に向けられている。この実施例では、データバスにスイッチを設けて演算をしながら転送できるようにする。この構成により、コプロセッサのレジスタ容量を増加させることなく、実行時間の短縮と電

流アタック対策の両方に効果的である。

【0053】コプロセッサ専用レジスタ（CDA、CDB、CDN、CDW）は、同図に示すように4つのレジスタがCPUとコプロセッサの演算器との間で排他的に使用されている。2ビット処理を行う場合、2回の $A^2 \bmod N$ を行いながらその間にBの値をRAMからコプロセッサ専用レジスタユニット中のBレジスタCDBに転送できるようにすると効率的である。

【0054】コプロセッサのAレジスタCDAとBレジスタCDBのI/Oを分け、それぞれにリード/ライトバッファ（R/W Buffer）を設けて、それぞれ独立に動作できるようにする。演算器が $A^2 \bmod N$ を演算している間は、制御信号によりデータバスをパス1（path1）につなぎ、図示しないCPUのRAMからBの値を上記独立に設けられたリード/ライトバッファを介してBレジスタCDBに転送する。次に演算器が $AB \bmod N$ を実行する際には、制御信号によりパス2（path2）に切り換え、上記BレジスタのB値を演算器に送り上記CPUがBレジスタCDBにアクセスできないようにする。この方法を取ると、 $A^2 \bmod N$ を演算動

作と、B値の転送動作が同時に行なわれるから演算時間が短縮されるだけでなく、演算と転送の消費電流が重なるため双方の波形が識別できなくなり、電流アタック対策に有効である。

【0055】図20には、この発明に係るICカード用チップの他の一実施例の要部ブロック図が示されている。この実施例では、暗号処理用演算ユニットとメモリ（RAM）間の転送の際、メモリにカウンタを設けるようにするものである。この実施例では、2ビット処理に用いる4通りの値、あるいは3ビット処理に用いる8通りの値をコプロセッサ外部メモリRAMからコプロセッサ専用レジスタユニット中のBレジスタCDBに転送する際の電流損乱を行なうようにするものである。

【0056】この実施例では、前記図2に示したようなICカード用チップにおいて、RAMの側にカウンタが設けられる。RAMは、カウンタで形成されたアドレス信号をデコードしてデータをデータバスに送出する。このとき、アドレスバスには、乱数発生器が形成された偽アドレスが送出される。これにより、アドレスとデータとの相関が無くなり、電流解析を困難とさせることができる。

【0057】図21には、上記カウンタの一実施例のブロック図が示されている。カウンタは、転送したいブロックの最初のアドレスを保持する先頭アドレスレジスタとインクリメンタを用い、ブロック転送をイネーブルにするイネーブル信号とクロック又はリード/ライト信号などによるインクリメント指示信号で制御する。ブロック転送を開始する際、まず転送の先頭アドレスと転送開始のイネーブル信号がCPUよりカウンタに送信され、上記先頭アドレスレジスタに保持される。その後は、インクリメント指示信号によって、インクリメンタが動作して先頭アドレスレジスタの先頭アドレス $A+1$ を形成して、アドレスを生成するとともに上記先頭アドレスレジスタの内容を書き換えるので、図22のタイミング図に示すように、RAMアドレスが順番にインクリメント A 、 $A+1$ 、 $A+2$ 、・・・されていき、そのアドレスに従って順次データ D_A 、 D_{A+1} 、 D_{A+2} 、・・・が書込まれ/読み出される。

【0058】この実施例では、ブロック転送がイネーブルになった後はアドレスバスからのアドレスをカウンタが受け付けられないため、アドレスバスにどのような値が来ようとデータは正しく読み出されていく。従って、アドレスバスに乱数発生器などで生成した乱数B、C、D、E・・・が出力されるとアドレスバスの消費電流を攪乱でき、この効果からチップ全体の消費電流を攪乱できるため、チップ内部動作の解析を困難にすることが可能になる。

【0059】図23には、この発明に係るICカード用チップの更に他の一実施例を示す要部ブロック図が示されている。この実施例でも、暗号処理用演算ユニットとメモリ(RAM)間の転送の際、メモリにカウンタを設けるようにするものであが、かかる暗号処理用演算ユニットとメモリRAMの最初のアドレスをも攪乱するようにアドレスオフセット機能が設けられる。つまり、乱数発生器などで生成した乱数をあらかじめCPUとカウンタ側に同時に転送しておき、ブロック転送の最初のアドレスに乱数を加えるか又は引くかした値をアドレスバスに出力する。カウンタ側ではアドレスバスの値を同じ乱数を用いて復号化し、最初のアドレスを得る。

【0060】図24には、上記転送動作を説明するためのタイミング図が示されている。乱数発生器で形成された乱数をあらかじめCPUとRAMに転送しておき、オフセット演算部1によりブロック転送の最初のアドレス*

- (1) input X, Y = $e_1, e_2, \dots, e_n, N, R$
- (2) $B = R^2 \bmod N$
- (3) $A = X$
- (4) $A = ABR^{-1} \bmod N + kN$
- (5) $B = A$
- (6) for $i = n-1$ to 1 step -1 {
- (7) $A = A^2 R^{-1} \bmod N + kN$
- (8) if $e_i = 1$ then $A = ABR^{-1} \bmod N + kN$
- (9) }
- (10) $A = AR^{-1} \bmod N + kN$
- (11) $A = A \bmod N$
- (12) output A

【0063】この発明の他の実施例では、前記図2コプロセッサ209において、上記アルゴリズム5のステップ(4)、ステップ(7)、ステップ(8)、ステップ(10)で示された「 $A = ABR^{-1} \bmod N + kN$ 」等に記述される「剰余乗算」を実行するようにされる。かかるコプロセッサ209は、後述するような演算回路と制御回路が含まれる。剰余乗算の入力値A、B、R、N及び出力値Aは専用レジスタ又はRAMなどの記憶装置に保持される。

【0064】図26には、この発明に用いられるコプロセッサの他の一実施例のブロック図が示されている。同図において33は第1の積和演算器、34は第2の積和演算器、35は一次記憶値Tempを保持するテンポラ

* Aに乱数Sを加えるか引くかしたアドレス $A \pm S$ をアドレスバスに送出する。カウンタ側では、アドレスバスの値を同じ乱数Sを用いて復号化し、オフセット演算部2により最初のアドレスAを得て、以後前記同様にインクリメントしてアドレス $A+1$ 、 $A+2$ ・・・を生成する。このようなアドレス $A+1$ 、 $A+2$ に同期して、乱数発生器が乱数B、C、D・・・をアドレスバスに送出するので、先頭のアドレスを含めてアドレスバスの消費電流を攪乱でき、チップ内部動作の解析をいっそう困難にすることが可能になる。

【0061】前記実施例のような暗号化/復号化装置において、べき乗剰余演算「 $X^Y \bmod N$ 」(X, Y, N は正の整数)を用いた場合、 X, Y, N が、通常100ビット〜2000ビット程度の非常に大きな数を使用されるため、「 $X^Y \bmod N$ 」をいかにして高速に実行するかが重要となる。その一つの解法として、剰余乗算「 $ABR^{-1} \bmod N$ 」を実行する次のようなアルゴリズムが知られており、本願出願人においては、特開平10-21057号公報(米国登録番号5,961,578)において、「 $ABR^{-1} \bmod N$ 」のアルゴリズムを基にした積和演算器を用いたマイクロコンピュータを提案している。

【0062】上記アルゴリズムは、次のステップ(1)ないし(12)からなる。

リレジスタ、36は値Aの格納に利用されるレジスタ、37は値Bの格納に利用されるレジスタ、38は値Nの格納に利用されるレジスタである。39はMi生成ロジック、40はMi生成ロジック39で生成された値M_iを保持するラッチ、41は「 $\div 2^i$ 」を行うためのシフト回路である。

【0065】この実施例では、前記公報に詳細に説明されているようなブロック分割に基づいて演算「 $(AB_i + M_i)/2^i$ 」を実行するようにされる。まず、第1の積和演算器33は、レジスタ35の値Temp、レジスタ36の値A、レジスタ37の値B、を入力として、積和演算「 $Temp + A \cdot B_i$ 」を実行する。その演算結果は値Temp2として次段の第2の積和演算器34

へ送られる。上記値Temp2は $n+L$ ビット長の整数である。

【0066】一方、Mi生成ロジック39は、Lビット長の数A、B、Nを入力としてLビットの整数M_iを生成し、この正数M_iはレジスタ40に一時的に保持される。第2の積和演算器34は、前記Temp2、N、M_iを入力として、積和演算「Temp2+M_i・N」を実行する。 $n+L$ ビット長の演算結果の下位Lビットは全て0であり、これをシフト41によって消去して(すなわち 2^L で割って)、nビット長の結果が値Tempとしてレジスタ35に送られて保持される。

【0067】以上の動作を n/L 回繰り返し実行すれば、演算「(A+B・N)/R」が実現できる。これによれば、nビットの整数Mをあらかじめ計算して保持する必要はなく、Lビット長のM_iのみを積和演算器33の計算中に求めてレジスタ40に保持すればよく、値Mの計算時間の削除、および値Mを保持する記憶手段の規模を縮小することができる。さらに、積和演算器33と積和演算器34を直列的に接続して連続的に動作させることにより、 $n+L$ ビット長の中間結果Temp2を一時的に保持する記憶手段を特別に設けることも必要なくなる。

【0068】レジスタ35~38を積和演算器33、34にバス43で接続される。したがって前記レジスタ35~38をRAM42で構成することができる。これにより、半導体チップ上のレジスタ面積の低減が可能となる。また、この構成においては、特にデータバス43によるデータ転送量が多いため、バス幅が大きくなって半導体チップの面積が大きくなるようにする必要が生じるが、図26の実施例のように積和演算器33と積和演算器34を直列的に接続することにより、中間結果Temp2をデータバスを用いて転送することが不要になるため、バスによるデータ転送量の低減を図ることができる。

【0069】この実施例のコプロセッサでは、第1の積和演算器33でTemp=0、第2の積和演算器34でM_i・N=0、さらにセレクト41による「 $\div 2^L$ 」の動作を行なわないことにより、同図に示される演算手段を、「A・B_i」のような多倍長乗算(小さな数B_iとその多倍長に相当する大きな数Aとの乗算)を実行する回路として使用することができる。「A・B_i」のような多倍長乗算演算は、上記アルゴリズムのステップ(2)の演算「Rⁱ mod N」をマイクロプロセッサ201を用いて実行するときに適用されることにより、その演算の高速化を図ることができる。

【0070】図27の「Rⁱ mod N」の計算の概念図に示されているように、 $R=2^n$ 、 $n=512$ とされ、Nは512ビット、Rⁱは最上位ビットだけが1で下位側1024ビット全てが0の値とされる。マイクロプロセッサで演算「Rⁱ mod N」を行うとき、大きな数の

Rⁱを同様に大きな数のNで直接に除算するのは効率的でないから、被除数を最上位側から64ビット単位のブロックとして把握し、また、除数を最上位側から32ビット単位のブロックとして把握し、順次上位側のブロック同士を対象に除算を行い、それによって得られる値を商の概数として把握する。

【0071】同図において、例えばQ(=Da÷Na)を商の概数として把握する。概略的には、Rⁱの上位側に対して「Q・Na」を減算し、その減算結果の上位側に対して「Q・Nb」を減算する。「Q・Nb」の減算結果に対して同様の処理を行い、更にその結果の対して同様の処理を繰り返すという手法によって、「Rⁱ mod N」の結果を得ることができる。

【0072】実際にはその途上で、余剰ビットを消去するための減算処理が介在される。このとき、前記演算「Q・Nb」の処理は、第1回目では32ビットと480ビットという大きな数の乗算処理とされる。しかもそのような大きな数の乗算処理は何回も繰り返される。このとき、前記図26に示されるコプロセッサによって演算可能な前記「A・B_i」のような多倍長乗算演算を利用することにより、換言すれば、そのような多倍長乗算演算をコプロセッサに負担させれば、上記アルゴリズム5におけるステップ2の演算「Rⁱ mod N」をマイクロプロセッサ201を用いて実行するとき、その演算処理の高速化を図ることができる。

【0073】前記のようなアルゴリズムにおける「A=ABR⁻¹ mod N」の演算処理では、前記公報(特開平10-21057号)において詳述されているように、剰余乗算において、オーバーフロー有りのときには更に演算結果WからNを減算W-Nするものであるため、オーバーフローの有無により演算時間や消費電流の違いが生じる。このため、前記のようなICカードLSIの消費電流を観測し、そのタイミングや統計的処理の結果からチップ内の動作を解析されてしまう可能性を持っている。

【0074】図28には、この発明に係る暗号化処理用演算ユニットの一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットは、前記のようなICカード等に搭載される1チップのマイクロコンピュータに含まれるコプロセッサに含まれる。

【0075】図28において、前記図26に示した第1と第2の積和演算器33、34を含む積和演算器により、前記A・R⁻¹ mod N又はABR⁻¹ mod Nの演算が行われ、その演算結果WはテンポラリレジスタTempに格納され、演算結果にオーバーフローが発生した場合には演算器からのオーバーフローフラグOVが制御論理のOV格納レジスタに記憶される。そして、続いて上記剰余乗算の後にテンポラリレジスタTempに格納された演算結果W-Nの減算が行われる。

【0076】上記オーバーフローフラグOVが有る時

(論理1)には上記減算 $W-N$ の結果は、テンポラリレジスタTempに格納され、オーバーフローフラグOVが無い時(論理0)には減算 $W-N$ の結果は、テンポラリレジスタTempに格納されず、上記テンポラリレジスタTemp以外の適当な記憶回路、例えばレジスタAに格納される。つまり、前記減算 $W-N$ と、それにより形成された無効データを適当な記憶回路に格納する動作は、前記錯乱目的のダミー動作とされる。これにより、*

```

W ← (AB+MN) / R
Store OV bit
if OV then
    W ← W-N (正規のオーバーフロー処理とWへの書き込み)
Else
    A ← W-N (ダミーのオーバーフロー処理とAへの書き込み)
Exchange W and A
Output A

```

【0078】上記プログラムにおいて、Wはテンポラリレジスタ及びそのデータを表している。そして、オーバーフローフラグOV無しの場合に、テンポラリレジスタTempのアドレスをレジスタAのアドレスを交換することにより、オーバーフローフラグOVの有/無しに対応してW又はAのデータが有効なデータとして出力さ*

```

W ← (AB+NM) / R
Store OV bit
A ← W-N (オーバーフロー処理とレジスタAへの書き込み)
if! OV then
    Exchange W and A
Else nop
Output A

```

【0080】つまり、オーバーフローOVの有無に無条件でのオーバーフロー処理のための減算 $W-N$ とその減算結果をAレジスタの書き込みを行った後に、オーバーフローフラグOVが無ければ、Exchange W and Aのようにアドレス交換を行ってレジスタAのアドレス指定によりテンポラリレジスタ(W)のデータ★

```

W ← (AB+NM) / R
Store OV bit
Exchange W and A
W ← A-N (オーバーフロー処理とレジスタAへの書き込み)
if OV then
    Exchange W and A
Else nop
Output A

```

【0082】つまり、オーバーフローOVの有無に無条件でのオーバーフロー処理のための減算 $W-A$ を行う前にExchange W and Aのようにアドレス交換を行って $A-N$ の減算、つまりは $W-N$ の減算を行ってテンポラリレジスタ(W)、つまりレジスタAにデータを出力させる。そして、オーバーフローフラグOVが有れば、Exchange W and Aのように

* 前記剰余乗算においてオーバーフローが生じないときでも、 $W-N$ の減算及びその演算結果をレジスタに格納することに伴うICカードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるようとなる。

【0077】上記の信号処理は、次のようなプログラムによって実施される。

※れる。この実施例では、Exchange W and Aのようなアドレス交換によって、レジスタAのアドレス指定によりテンポラリレジスタ(W)のデータを出力させるものである。

【0079】上記の信号処理は、次のようなプログラムに置き換えることができる。

30★を出力させ、無ければアドレスを交換することなくAレジスタのデータ $W-N$ を有効なデータとして出力させる。

【0081】上記の信号処理は、更に次のようなプログラムに置き換えることができる。

再度アドレス交換を行い、レジスタAのアドレス指定によりレジスタAのデータを、オーバーフローフラグOVが無ければ前記のように交換したままレジスタAのアドレス指定によりテンポラリレジスタ(W)のデータを出力させる。この構成では、レジスタへの書き込みを行う論理回路は、みかけ上テンポラリレジスタ(W)を書き込むような構成となり、レジスタAへの書き込み用論理

が不要となって回路の簡素化が可能になる。

【0083】前記テンポラリレジスタTempとレジスタAのアドレス交換は、フラグ反転回路により実現できる。つまり、アドレスバスから供給されるアドレス信号のうち、例えば最下位ビットのような1ビットがテンポラリレジスタTempとレジスタAとで異なるように設定しておき、フラグ反転回路により選択的にかかるビットを交換するだけで、テンポラリレジスタTempに割り当てられたアドレス指定によりレジスタAを選択でき、逆にレジスタAに割り当てられたアドレスによりテンポラリレジスタTempを選択することができる。

【0084】図28の実施例において、2つのレジスタTempとAを用い、オーバーフローフラグOVに対応して、常に一方（例えばテンポラリレジスタTemp）に有効データを格納させ、前記のようなアドレス交換によってレジスタAを指定するアドレスにより有効なデータを出力させる。前記減算W-Nと、それにより形成された無効データを適当な記憶回路に格納する動作が錯乱目的のダミー動作とされる。これにより、前記剰余乗算においてオーバーフローが生じないときでも、W-Nの減算及びその演算結果をレジスタに格納することに伴うICカードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるものとなる。

【0085】前記実施例のように積和演算器の前記のようなオーバーフローフラグに代えてボロー（Borrow）フラグBRを利用するものであってもよい。つまり、前記 $A \leftarrow R \cdot \text{mod } N$ 又は $ABR \cdot \text{mod } N$ の演算結果WはテンポラリレジスタTempに格納し、W-Nの減算が行われときの演算器からのボローフラグBRを記憶し、ボローフラグBRが有るときのみ、テンポラリレジスタTempとレジスタAのアドレスを交換し、最終的にはレジスタAのアドレス指定により有効なデータを読み出すようにするものであってもよい。

【0086】上記の信号処理は、次のようなプログラムにより実現できる。

```

W ← (AB+NM) / R
A ← W-N
Store BR bit
if BR then
    Exchange W and A
Else nop
Output A

```

【0087】図29には、この発明に係る暗号化処理用*

```

W ← (AB+NM) / R
Store OV bit
if OV then
    A ← W-N （正規のオーバーフロー処理とAへの書き込み）
Else
    X ← W-N （ダミーのオーバーフロー処理とXへの書き込み）

```

* 演算ユニットの他の一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットも前記のようなICカード等に搭載される1チップのマイクロコンピュータに含まれるコプロセッサに含まれる。この実施例では、データバスの信号と前記のような積和演算器の出力とのうちのいずれか一方を前記のようなOVフラグ格納レジスタに記憶されたオーバーフローフラグOVに従って出力させるセレクトが追加される。

【0088】上記剰余乗算の後W-Nの減算が行われ、この減算結果W-Nと演算のためにデータバス上に読み出されたWの値がセレクトに入力され、オーバーフローフラグOVが有るときには減算結果W-Nが、オーバーフローフラグOVが無いときにはデータバスWの値が選択され、この選択された値がレジスタAに格納され、Aが最終的に有効なデータとして出力されることにより、W-Nの減算とレジスタへの書き込みに伴うICカードやマイクロコンピュータの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とされる。

【0089】図30には、この発明に係る暗号化処理用演算ユニットの更に他の一実施例の要部ブロック図が示されている。この実施例は、前記図28の実施例においてレジスタブロックにレジスタXが追加される。前記同様に剰余乗算の後W-Nの減算が行われ、オーバーフローフラグOVが有る時にはこの減算結果W-NがレジスタAに、オーバーフローフラグOVが無い時にはダミー演算専用のレジスタXに減算W-Nが書込まれる。この後に、オーバーフローフラグOVが無い時には、テンポラリレジスタ（W）とレジスタAとのアドレスを交換して、最終的にレジスタAを選択するアドレスにより有効なデータを出力させる。

【0090】上記の信号処理は、次のようなプログラムにより実現できる。

Exchange W and A Output A

【0091】上記の実施例から得られる作用効果は、下記の通りである。すなわち、

(1) 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むICカードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0092】(2) 上記に加えて、上記暗号化処理又は復号化処理は、RSA暗号法などに応用可能なべき乗剰余乗算動作を含むようにすることにより、機密保護の強化を実現したICカードを得ることができるという効果が得られる。

【0093】(3) 上記に加えて、上記べき乗剰余乗算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行わせることにより、高速なデータ処理を行なうようにすることができるという効果が得られる。

【0094】(4) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A' \bmod N$ と $A=AB \bmod N$ の演算を交互行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A' \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A' \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理0のときの $A=AB \bmod N$ の演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0095】(5) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタからなるレジスタブロックを用い、上記Yの特定ビット e_i の論理1又は0によってゲート回路を制御し、所定のレジスタに供給されるライトストロブ信号の伝達を制御して、上記演算結果のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0096】(6) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックを用い、上記Yの特定ビット e_i の論理1又は0によってゲート回路を制御し、上記リードライトバッファに供給されるライトストロブ信号の伝達を制御して、上記演算結果

のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0097】(7) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブロックを用い、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間セクタを設けて上記Yの特定ビット e_i の論理1又は0によって制御して、リードライトバッファに書き込まれた演算結果のうち有効なデータを所定のレジスタに格納し、無効なデータが上記ダミーレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【0098】(8) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A' \bmod N$ と $A=AB \bmod N$ の演算を交互行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A' \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A' \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A' \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A' \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A' \bmod N$ の演算が開始されるまでの間も上記 $A=AB \bmod N$ の動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【0099】(9) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A' \bmod N$ と $A=AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A' \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A' \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【0100】(10) 外部端子がリードライト装置と

電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードに、上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にしたICカードを得ることができるという効果が得られる。

【0101】(11) 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードに、上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にしたICカードを得ることができるという効果が得られる。

【0102】(12) 暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができるという効果が得られる。

【0103】(13) 上記に加えて、上記マイクロコンピュータのモジュール構成を1つの半導体基板上において形成することにより、小型化を図りつつ電流波形以外の直接的なプログラム又はデータ等の解読も防止することができるという効果が得られる。

【0104】(14) 上記に加えて、上記マイクロコンピュータの暗号化処理又は復号化処理を、RSA暗号法などに応用可能なべき乗剰乗算動作を含むものとし、上記べき乗剰乗算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なうようにすることにより、高速な暗号処理動作を行なうことができるという効果が得られる。

【0105】(15) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理0のときの $A=AB \bmod N$ の演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0106】(16) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A^2 \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A^2 \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A=AB \bmod N$ の動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0107】(17) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=X^2 \bmod N$ と $A=AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算を行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A=AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0107】(18) 上記に加えて、上記暗号化処理用演算ユニットにより、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、Yのビットの値に応じて、 $A=A^2 R^{-1} \bmod N$ 、 $A=AB R^{-1} \bmod N$ の演算を行うとともに、演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算 $W-N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W-N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0108】(19) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰乗算の後に上記第1記憶回路の演算結果WからNの減算 $W-N$ を行い、その演算結果を上

記オーバーフローフラグOVが有る時には上記第1の記憶回路に格納し、オーバーフローフラグOVが無い時には上記第1記憶回路とは異なる第2の記憶回路に上記錯乱目的のダミー動作として格納し、上記第1の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した読解を困難にすることができるという効果が得られる。

【0109】(20) 上記に加えて、上記 $A^i R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算W-Nを行い、オーバーフローフラグOVが有るときに上記演算結果W-Nをセレクトにより選択され、オーバーフローフラグOVが無いときには上記第1記憶回路の演算結果Wをセレクトにより選択して第2の記憶回路に格納することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した読解を困難にすることができるという効果が得られる。

【0110】(21) 上記に加えて、上記 $A^i R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算W-Nを行い、オーバーフローフラグOVが有るときには減算W-Nを第2の記憶回路に記憶し、オーバーフローフラグOVが無いときには減算W-Nを第3の記憶回路に記憶し、オーバーフローフラグOVが有るときには上記第2の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグOVが無いときには上記第1の記憶回路のデータが有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した読解を困難にすることができるという効果が得られる。

【0111】(22) 上記に加えて、上記 $A^i R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果WからNの減算結果W-Nを第2の記憶回路に格納し、オーバーフローフラグOVが無いとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するアドレスにより第1の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグOVが有るとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして第2の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した読解を困難にす

とができるという効果が得られる。

【0112】(23) 上記に加えて、上記 $A^i R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路と第2の記憶回路のアドレスを交換し、第2の記憶回路を選択するアドレスにより選択される演算結果値WからNの減算W-Nが行われて第1の記憶回路を選択するアドレスにより選択される第2の記憶回路に減算結果W-Nを格納し、オーバーフローフラグOVが有るときにのみ上記アドレスを再度交換し、第1の記憶回路を選択するアドレスにより選択される第1又は第2の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した読解を困難にすることができるという効果が得られる。

【0113】(24) 上記に加えて、上記 $A^i R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、上記剰余乗算の後に上記第1の記憶回路の演算結果値WからNの減算W-Nを行って第2の記憶回路に格納し、このW-Nの減算が行われた時の演算器のポロフフラグBRを記憶し、ポロフフラグBRが有るときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するアドレスにより第1の記憶回路の演算結果Wを出力し、ポロフフラグBRが無いときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして、上記第2の記憶回路を選択するアドレスにより第2の記憶回路の演算結果W-Nを出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した読解を困難にすることができるという効果が得られる。

【0114】(25) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットとして、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、Yのビットの値に応じて、 $A=A^i R^{-1} \bmod N$ 、 $A=ABR^{-1} \bmod N$ の演算を行い、演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算W-Nを行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算W-Nに対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した読解を困難にすることができるという効果が得られる。

【0115】(26) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記

$A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第1の記憶回路に格納し、演算器からのオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1記憶回路の演算結果 W から N の減算 $W-N$ を行ない、演算結果を上記オーバーフローフラグOVが有る時には上記第1の記憶回路に格納し、オーバーフローフラグOVが無い時には上記第1記憶回路とは異なる第2の記憶回路に上記錯誤目的のダミー動作として書き込み、上記第1の記憶回路の演算結果が有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0116】(27) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果 W から N の減算 $W-N$ を行ってオーバーフローフラグOVが有るときにセクタにより上記演算結果 $W-N$ を選択し、オーバーフローフラグOVが無いときにはセクタにより上記第1記憶回路の演算結果 W を選択して第2の記憶回路に格納して有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0117】(28) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果 W から N の減算 $W-N$ を行ってオーバーフローフラグOVが有るときには減算結果 $W-N$ を第2の記憶回路に記憶し、オーバーフローフラグOVが無いときには減算結果 $W-N$ を第3の記憶回路に記憶し、オーバーフローフラグOVが有るときには上記第2の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグOVが無いときには上記第1の記憶回路のデータを有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0118】(29) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路の演算結果 W から N の減算結果 $W-N$ を第2の記憶回路に格納し、オーバーフローフラグOVが無いとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するア

ドレスにより第1の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグOVが有るとき第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして第2の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0119】(30) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1の記憶回路と第2の記憶回路のアドレスを交換し、第2の記憶回路を選択するアドレスにより選択される演算結果値 W から N の減算 $W-N$ を行って第1の記憶回路を選択するアドレスにより選択される第2の記憶回路に減算結果 $W-N$ を格納し、オーバーフローフラグOVが有るときにのみ上記アドレスを再度交換し、第1の記憶回路を選択するアドレスにより選択される第1又は第2の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0120】(31) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、上記剰余乗算の後に上記第1の記憶回路の演算結果値 W から N の減算 $W-N$ を行って第2の記憶回路に格納し、この $W-N$ の減算が行われた時の演算器からポロフフラグBRを記憶し、ポロフフラグBRが有るときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスを逆転させて、上記第2の記憶回路を選択するアドレスにより第1の記憶回路の演算結果 W を出力し、ポロフフラグBRが無いときには、第1の記憶回路と第2の記憶回路を選択する最下位アドレスをそのままにして、上記第2の記憶回路を選択するアドレスにより第2の記憶回路の演算結果 $W-N$ を出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0121】以上本発明者よりなされた発明を実施例に基づき具体的に説明したが、本願発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、ICカードには、1つの半導体集積回路装置を搭載するもの他、複数の半導体集積回路装置が搭載されるものであってもよい。マイクロコンピュータは、1つの半導体集積

回路装置に形成されるもの他、CPUとその周辺回路が複数チップで構成されて、1つのモジュール基板上に搭載されてなるものであってもよい。

【0122】演算処理は前記のような暗号処理を行なうべき乗剰余乗算法の他に、図25図に示したフローチャート図のように演算Aと演算Bを持ち、演算Aの結果により演算Bを行なうか否かの分岐を持つような演算処理、あるいは演算動作でのオーバーフローの有無に対応して、次の演算処理が選択的に追加される場合に等に広く利用することができる。つまり、演算Aの次に演算Bを実行し、演算Aの結果から演算Bが不要なら、その演算結果を無効にするような演算処理を行なえば、前記のような暗号処理以外の機密動作を必要とするデータ処理のハッキング対策として有益なものとなる。

【0123】上記マイクロコンピュータは、データ処理装置とかかるデータ処理装置によるデータ処理手順が書き込まれたROMを含んで記データ処理手順に従ってデータの入出力動作が行われるものであれば何であっててもよい。例えば、前記のようなICカード用チップの他に、ゲーム用等の1チップマイクロコンピュータ等のように機密保護の必要な各種マイクロコンピュータに広く適用できるものである。この発明は、機密保護を必要とする各種ICカード及びマイクロコンピュータに広く利用できる。

【0124】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むICカードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができる。

【0125】暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができる。

【図面の簡単な説明】

【図1】この発明が適用されるICカードの一実施例を示す外観図である。

【図2】この発明に係るICカードに搭載されるICカード用チップの一実施例を示す概略ブロック図である。

【図3】この発明に係るコプロセッサの一実施例の動作を説明するためのタイミング図である。

【図4】図3のコプロセッサの動作を説明するためのフ

ローチャート図である。

【図5】図3のコプロセッサの一実施例を示すブロック図である。

【図6】図3に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図7】図3のコプロセッサの他の一実施例を示すブロック図である。

【図8】図3のコプロセッサの他の一実施例を示すブロック図である。

10 【図9】この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図である。

【図10】図9に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図11】この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図である。

【図12】この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図である。

【図13】この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図である。

20 【図14】図11ないし図13に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図15】この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図である。

【図16】この発明に係るコプロセッサの演算動作の他の一実施例を示すフローチャート図である。

【図17】この発明に係るコプロセッサの他の一実施例を示すブロック図である。

30 【図18】この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図19】この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図20】この発明に係るICカード用チップの他の一実施例を示す要部ブロック図である。

【図21】図20のカウンタの一実施例を示すブロック図である。

【図22】図20のICカード用チップの動作の一例を示すタイミング図である。

40 【図23】この発明に係るICカード用チップの更に他の一実施例を示す要部ブロック図である。

【図24】図23のICカード用チップの動作の一例を示すタイミング図である。

【図25】この発明が適用可能な演算動作を説明するためのフローチャート図である。

【図26】この発明に用いられるコプロセッサの他の一実施例を示すブロック図である。

【図27】この発明における「 $R^i \bmod N$ 」の計算方法を示す概念図である。

50 【図28】この発明に係る暗号化処理用演算ユニットの一実施例を示す要部ブロック図である。

【図29】この発明に係る暗号化処理用演算ユニットの他の一実施例を示す要部ブロック図である。

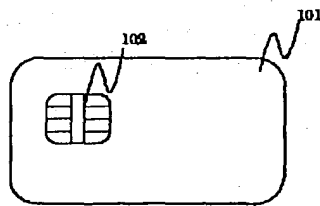
【図30】この発明に係る暗号化処理用演算ユニットの更に他の一実施例を示す要部ブロック図である。

【符号の説明】

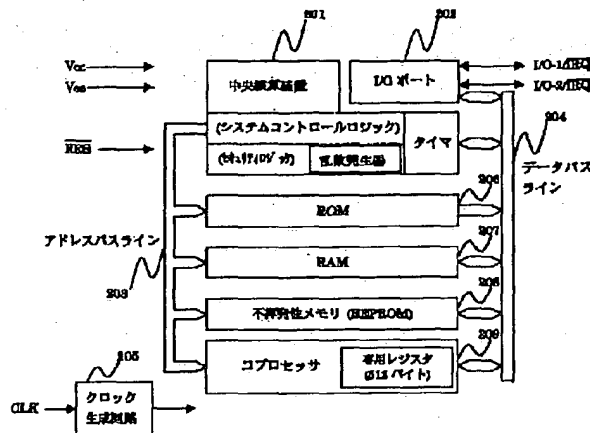
201…中央処理装置(CPU)、202…I/Oポート、203…アドレスバス、204…データバス、205…クロック生成回路、206…ROM、207…RAM、208…EEPROM、209…コプロセッサ(暗号化処理用演算ユニット)、CDA、CDB、CDN、CDW…レジスタ、33、34…積和演算器、35…テンポラリレジスタ、36~38…レジスタ、39…Mi生成ロジック、40…Miを保持するラッチ(レジスタ)、41…シフタ、42…RAM、43…データバス。

*M、208…EEPROM、209…コプロセッサ(暗号化処理用演算ユニット)、CDA、CDB、CDN、CDW…レジスタ、33、34…積和演算器、35…テンポラリレジスタ、36~38…レジスタ、39…Mi生成ロジック、40…Miを保持するラッチ(レジスタ)、41…シフタ、42…RAM、43…データバス。

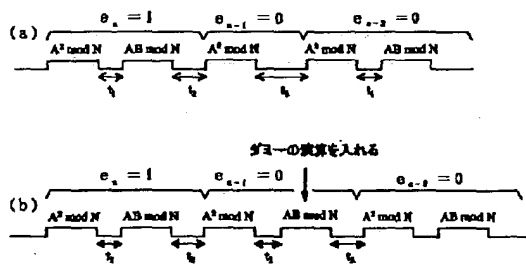
【図1】



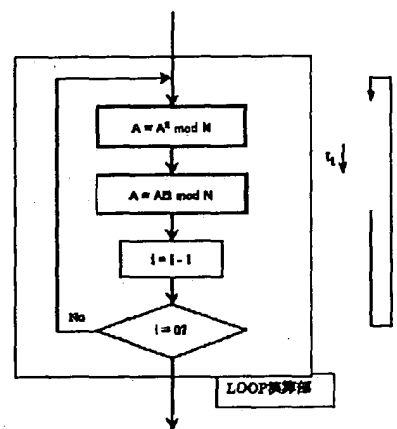
【図2】



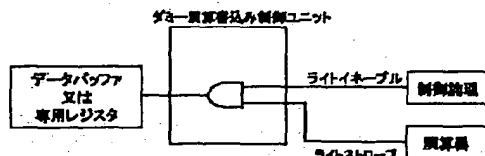
【図3】



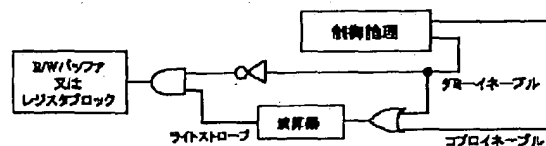
【図4】



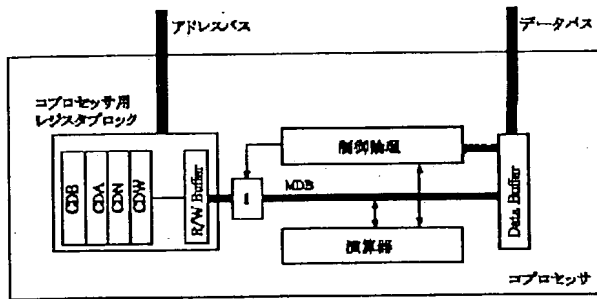
【図7】



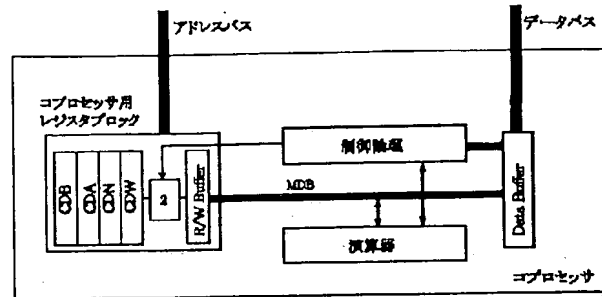
【図10】



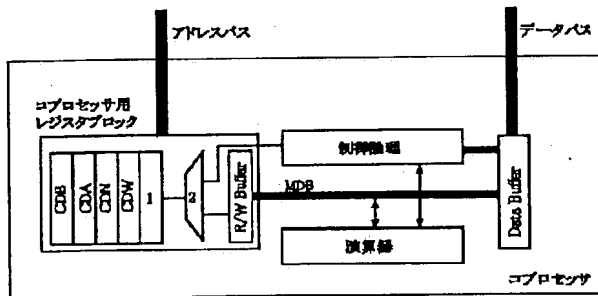
【図5】



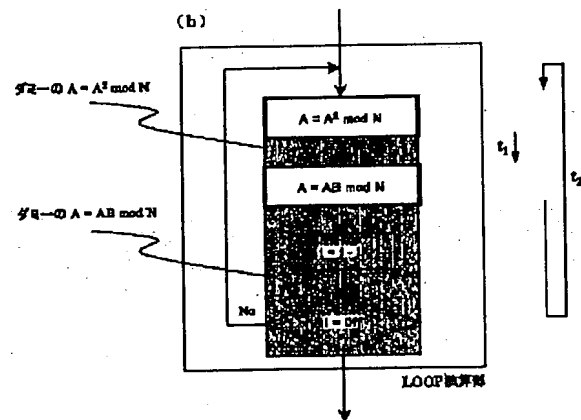
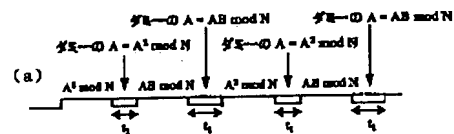
【図6】



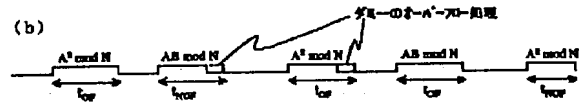
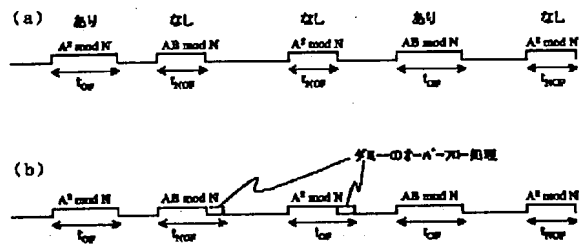
【図8】



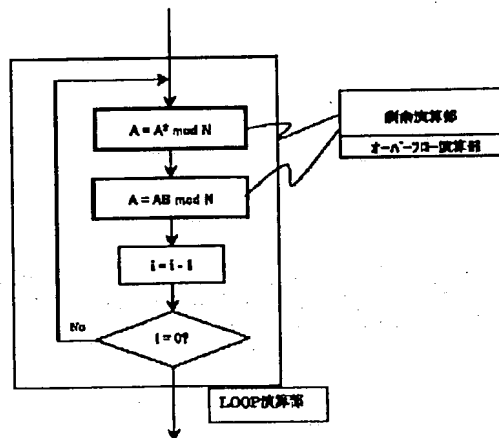
【図9】



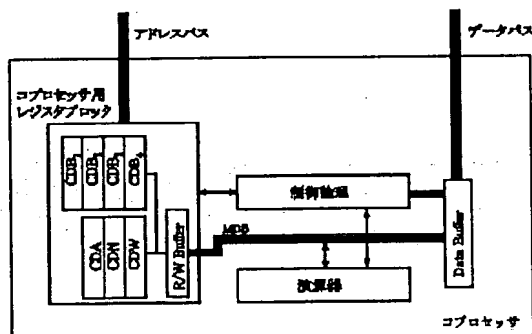
【図11】



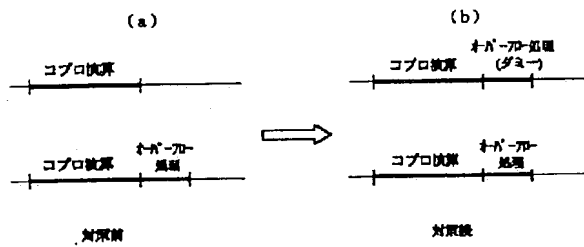
【図12】



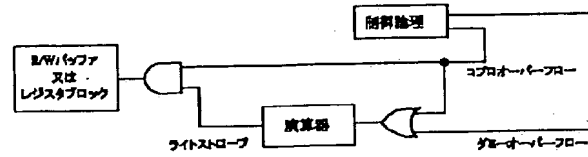
【図17】



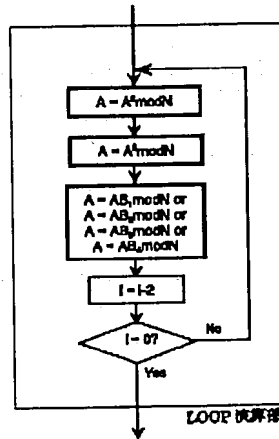
【図13】



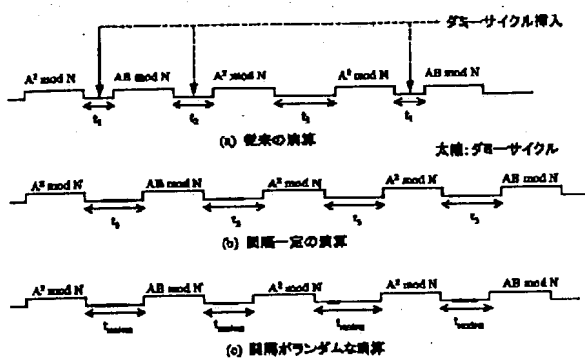
【図14】



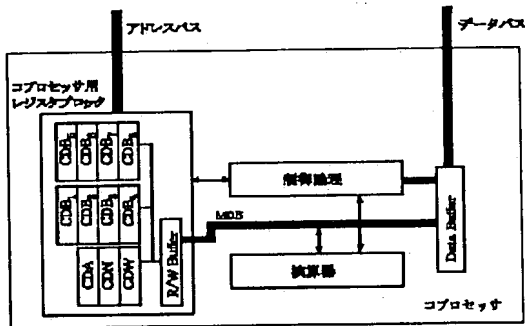
【図16】



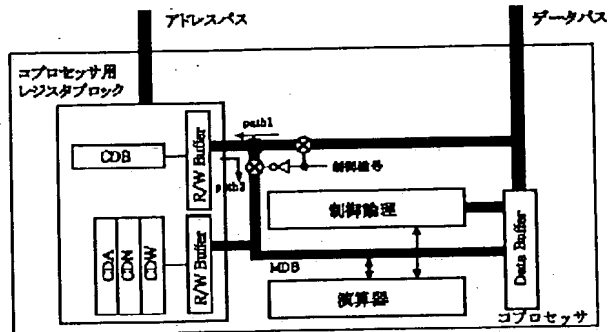
【図15】



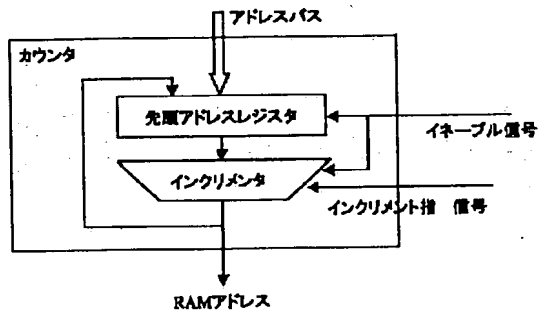
【図18】



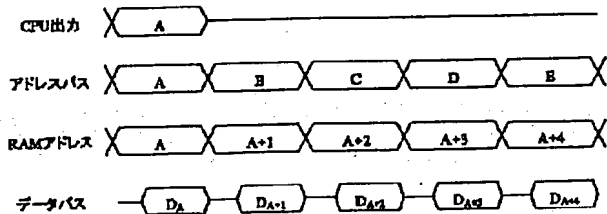
【図19】



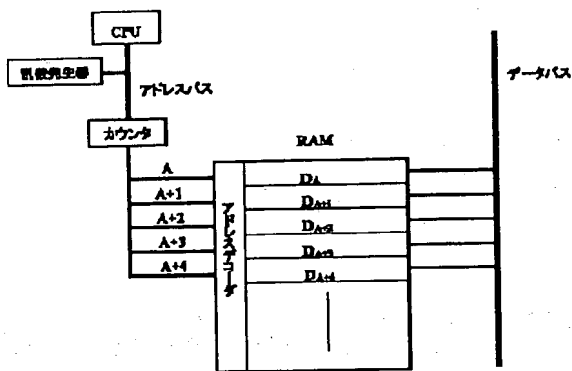
【図21】



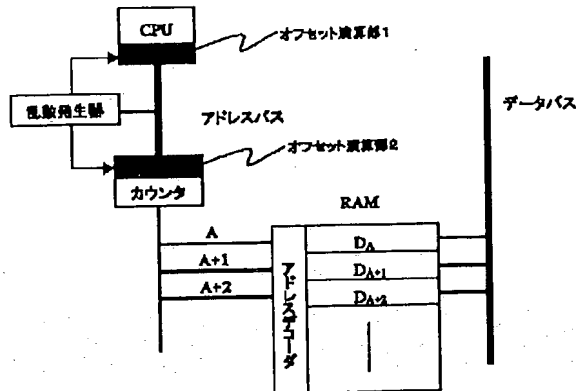
【図22】



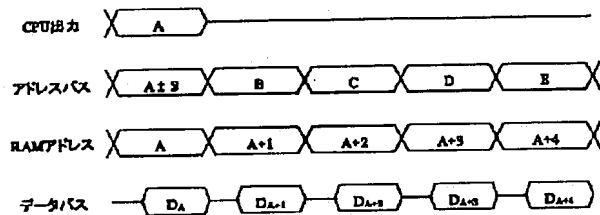
【図20】



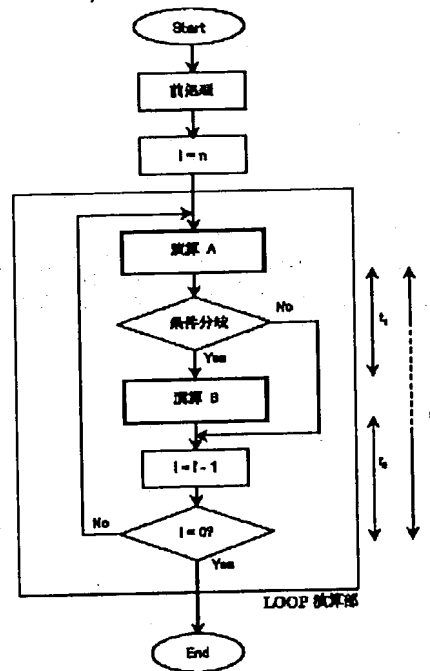
【図23】



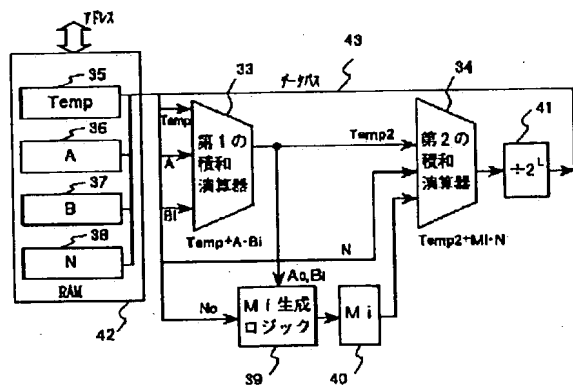
【図24】



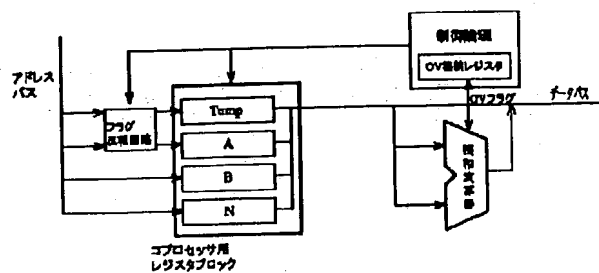
【図25】



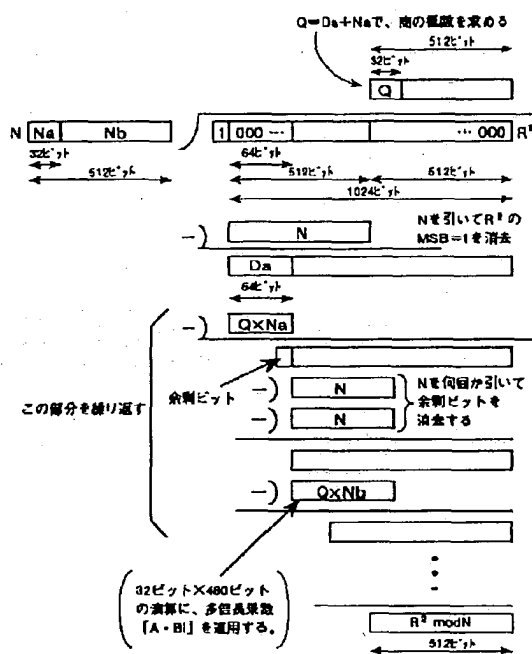
【図26】



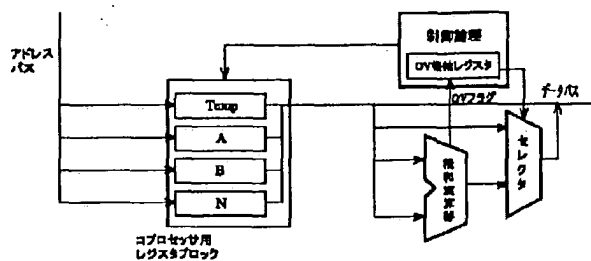
【図28】



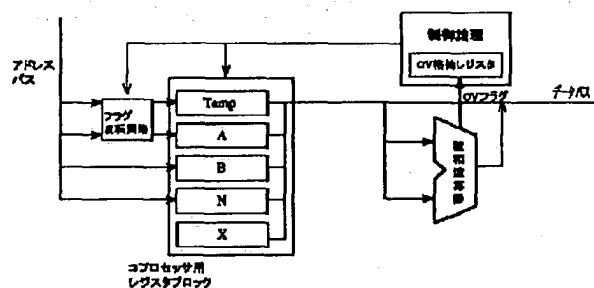
【図27】



【図29】



【図30】



フロントページの続き

- (72)発明者 中田 邦彦
東京都小平市上水本町五丁目20番1号 株
式会社日立製作所半導体グループ内
- (72)発明者 成吉 雄一郎
東京都小平市上水本町5丁目22番1号 日
立超エル・エス・アイ・システムズ内
- (72)発明者 塚元 卓
東京都小平市上水本町5丁目22番1号 日
立超エル・エス・アイ・システムズ内

- (72)発明者 平林 茂雄
東京都小平市上水本町5丁目22番1号 日
立超エル・エス・アイ・システムズ内
- (72)発明者 渡瀬 弘
東京都小平市上水本町5丁目22番1号 日
立超エル・エス・アイ・システムズ内
- (72)発明者 ▲高▼橋 雅聡
東京都小平市上水本町5丁目22番1号 日
立超エル・エス・アイ・システムズ内

Fターム(参考) 5B035 BB09 CA38
5B062 AA07 CC01 EE01 EE02 EE10
JJ10
5J104 AA16 AA47 EA04 NA07 NA18
NA22 NA35 NA37 NA40